The Unbearable Lightness of Monitoring:

Chothia, Tom; Cova, Marco; Novakovic, Christopher; Toro, C

DOI:
10.1007/978-3-642-36883-7_12

The Unbearable Lightness of Monitoring: Direct Monitoring in BitTorrent

Tom Chothia, Marco Cova, Chris Novakovic, and Camilo González Toro

School of Computer Science, University of Birmingham, UK

Abstract. It is known that BitTorrent file-sharing traffic is analysed to identify exchangers of copyrighted material. In general, copyright holders can perform monitoring using two approaches: *indirect monitoring*, where indirect clues of the sharing activity of a peer are considered (e.g., its presence in the peer list of a tracker), and *direct monitoring*, which establishes connections with peers to estimate their participation in sharing activity. Previous research has focused exclusively on indirect monitoring. We provide a broader characterisation of the monitoring of BitTorrent activity by considering both indirect and direct monitoring. In particular, we review previous work on indirect monitoring, provide features to detect peers engaged in such monitoring, and apply them to identify a number of monitoring organisations. Additionally, we introduce features that detect direct monitors, and provide the first ever measurements of direct monitoring, showing that it is now occurring.

Key words: BitTorrent, P2P monitoring, copyright enforcement

1 Introduction

BitTorrent is a decentralised peer-to-peer (P2P) protocol designed for the efficient transfer of large files. It is used by millions of users, contributing significantly to the volume of global Internet traffic [19]. BitTorrent users exchange a range of legal content: many Linux distributions rely on BitTorrent as a content delivery mechanism, and video game companies use it to provide updates and patches to their customers [2]. However, BitTorrent is also widely used (overwhelmingly so, according to one study [11]) for the illegal exchange of copyrighted material, such as music, movies and software.

Many copyright holders perceive this illegal exchange of content as a threat to their business models and have increasingly sought to prevent it. In particular, copyright holders are known to routinely monitor file-sharers, collect evidence of infringement, issue cease-and-desist letters and, in some cases, demand financial compensation from the users they deem to have infringed their copyright [8]. The task of policing BitTorrent is often outsourced to specialist *copyright enforcement agencies*.

One key aspect of BitTorrent monitoring is the precise set of techniques employed by enforcement agencies, which have never been disclosed publicly; in fact, the companies involved appear keen to avoid having their evidence being

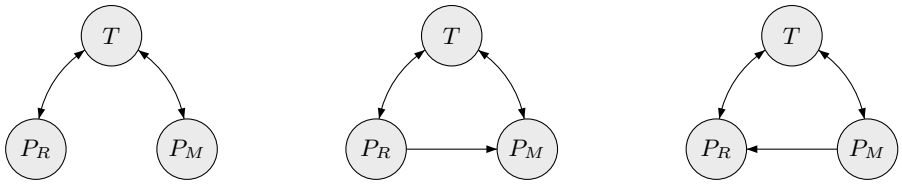

Fig. 1. Different methods by which a monitoring peer (P_M) may monitor a regular peer (P_R) via a tracker (T). From left to right: indirect monitoring; passive direct monitoring; active direct monitoring.

examined in court [8]. Nevertheless, two general approaches are possible: *indirect* and *direct* monitoring [17].

With indirect monitoring, enforcement agencies rely on indirect clues that a peer is uploading or downloading some content (i.e., by the presence of the peer's IP address in the group, or *swarm*, of peers reported by a BitTorrent *tracker* to be sharing the file — see Figure 1). A 2008 study by Piatek et al. [17] showed that indirect monitoring was extensively used by enforcement agencies. The study also demonstrated the high rate of false positives caused by this approach by implicating innocent devices such as printers and wireless access points as file-sharers, which later received cease-and-desist letters. More recent studies have confirmed that these flawed practices continue to be used [6, 18].

With direct monitoring, enforcement agencies collect first-hand evidence of a peer's activity. Direct monitoring can be *active* if the monitor establishes connections with peers to confirm that they are sharing a file, or *passive* if the monitor advertises its IP address to a tracker and waits for peers to connect to it (see Figure 1). Clearly, direct monitoring techniques have the potential to gather more conclusive evidence, but are also costlier (in terms of bandwidth and computational resources) when compared with indirect techniques; methods of improving the efficiency of direct monitoring have been proposed [1]. Documents recently filed in a New York Southern District Court case imply that at least one copyright enforcement agency is using some form of direct monitoring to collect its evidence against file-sharers [15]; however, at this time it is not clear whether comprehensive direct monitoring is in widespread use.

The goal of this work is to characterise the current state of BitTorrent monitoring by investigating it from several points of view. Firstly, we review indirect monitoring and assess various features to detect peers that are engaged in this activity (*how can indirect monitoring be detected?*). Secondly, we focus on direct monitoring and study its characteristics. The occurrence of this type of monitoring has not been studied before; thus, we want to introduce features to detect peers engaging in direct monitoring (*how can direct monitoring be detected?*), as well as investigate its mechanics (*how is direct monitoring performed?*). Thirdly, we assess whether the information gathered by monitoring agencies is accurate and conclusive (*what information is really collected?*). Finally, we investigate how users can defend themselves against monitoring.

We conducted this study by measuring the activity of 1,033 swarms across 421 trackers for 36 days over 2 years, collecting over 150GB of BitTorrent traffic. We note that our aim is to design and test novel monitoring detection techniques, rather than provide a comprehensive picture of BitTorrent monitoring.

The main contributions of this study are:

- We determine that indirect monitoring is still in use against BitTorrent users and devise more effective techniques to detect peers engaging in it;
- We find indications that certain entities engage in direct monitoring of BitTorrent users and provide features to detect such peers;
- We also notice that direct monitoring, in its current form, falls short of providing conclusive evidence of copyright infringement.

1.1 Related Work

A number of studies have focused on measuring and characterising specific properties of BitTorrent (e.g., [5,7]); other work has introduced improvements to the measuring process itself (e.g., [24, 26, 27]). The limitations of the evidence collected through indirect monitoring for legal cases motivated Bauer et al. [1] to design BitStalker, an active probing mechanism for identifying hosts using BitTorrent to download files. Wolchok and Halderman [25] have shown that BitTorrent's distributed hash tables can be quickly crawled to more efficiently monitor users' activity. Similarly, Le Blond et al. [12, 13] have demonstrated how protocol features can be leveraged for efficient spying on large numbers of BitTorrent users. While some of the techniques proposed in these papers are related to our work, our aims are quite different; rather than measuring the behaviour of the typical BitTorrent user, we wish to determine if and how monitoring is taking place by measuring the atypical behaviour of monitors.

The issue of detecting and understanding how the indirect monitoring of users' activity is performed on BitTorrent has received attention in the past. In a 2008 study, Piatek et al. [17] provided empirical evidence that enforcement agencies resort to indirect monitoring for identifying infringing users. They questioned the robustness of evidence collected via indirect monitoring and presented attacks that may cause arbitrary network users to be wrongly accused of infringement. Siganos et al. [20] described a set of network-level features that can be used for automatically detecting "deviant" clients, some of which are deemed to be indirect monitors. We revisit the issue of identifying indirect monitors and introduce a new and novel detection method; we show that our method is simple to compute and provides more accurate results than those of Siganos et al. [20] by ruling out false positives due to network address translation (NAT). We are the first to study whether direct monitoring is used by copyright enforcement agencies to identify file-sharers, and discuss techniques for detecting direct monitors.

A common approach to BitTorrent monitor evasion is to prevent interaction with peers that are suspected of monitoring at the transport layer (lists of suspicious peers are often referred to as *blocklists*). Potharaju et al. [18] offer

a blocklist generation technique for BitTorrent based on peers' participation in multiple swarms sharing the same content, arguing that simultaneously downloading multiple copies of the same content is suspicious. The blocklist approach only prevents direct monitoring and it is only effective if reliable techniques exist for identifying monitors. We compare our results with the contents of a popular blocklist and discover a high incidence of false positives and false negatives in the blocklist we examine.

1.2 Ethical Statement

The tension between BitTorrent users and copyright enforcement agencies is often described as an arms race [17, 20, 25], in which one side attempts to share content and the other attempts to monitor and disrupt this activity. As with previous studies in this area, we do not take a side in this arms race: the results we present could benefit both users (e.g., by improving the detection and blocking of monitors) and copyright enforcement agencies (e.g., by improving monitoring techniques). Furthermore, it has been noted previously [18] that the monitoring process used by copyright enforcement agencies may wrongly implicate researchers performing experiments in BitTorrent swarms. The features we present may enable them to design more conservative research experiments or to better interpret their results.

There are significant privacy concerns when reporting on data collected from BitTorrent traffic. To protect the privacy of the peers we monitored, we do not disclose the IP addresses of individual peers, and the peer lists and peer/peer communication data that were collected during monitoring will be destroyed when they are no longer required. The web addresses of notable trackers are revealed, but since they regularly track hundreds of thousands of torrents simultaneously, this poses no risk of a privacy violation. We only disclose the identity of copyright enforcement agencies that have publicly announced that they are monitoring BitTorrent. Following previous work in this area (e.g., [18]), we indicate Autonomous Systems (ASes) that appear to host large numbers of monitors, but we do not disclose individual ranges inside an AS.

Finally, in all of our data collection processes, we were careful not to upload or download any shared files; therefore, we have not participated in any copyright-infringing activity as a result of this study. Piatek et al. [17] deliberately implicated innocent network devices (such as printers and routers) in file-sharing to draw unsubstantiated cease-and-desist letters from copyright enforcement agencies; since their study was designed to highlight the shortcomings of indirect monitoring, and ours involved communicating directly with other peers from network devices potentially capable of infringing copyright, we did not design our study in a way that would intentionally cause us to receive cease-and-desist letters.

2 Background

Firstly, we provide an overview of the BitTorrent protocol, emphasising the aspects of the protocol that are relevant to our work. We focus on the original specification of the protocol.

2.1 Protocol Overview and Terminology

The BitTorrent protocol was designed to replace the distribution of large files via other, less efficient protocols, such as HTTP and FTP.

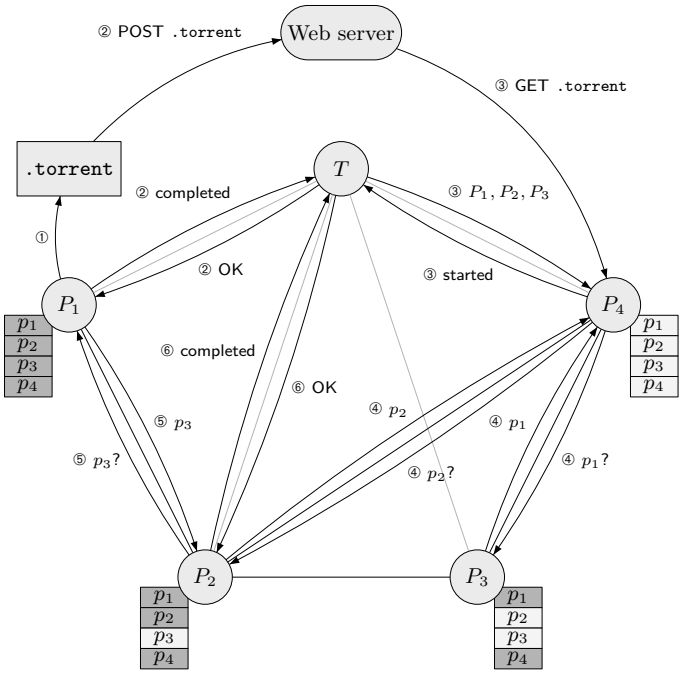

Fig. 2. A file being shared using BitTorrent.

Figure 2 summarises how a file is shared using BitTorrent. The user holding the file creates a *torrent file* containing metadata about the shared file. The shared file is described in terms of smaller *pieces*, which are divided further into *blocks*. When concatenated, the pieces produce the original shared file. The torrent file also contains the URL of a *tracker*, a centralised server that tracks which peers are downloading and uploading the shared file. A SHA-1 hash — the torrent's *infohash* — is used in all subsequent peer/tracker communication to uniquely identify this torrent. The torrent file can then be published (e.g., on a web server).

Users interested in the shared file download the torrent file and report their presence to the tracker by *announcing* to it — thus they become peers, and join the collective *swarm* of peers uploading and downloading pieces of the shared file. The tracker responds with a list of up to 200 IP addresses and port numbers of other peers in the swarm. Peers that hold a complete copy of the file are *seeders*, and those that do not are *leechers*.

Peers contact other peers in the swarm using the list of IP addresses given to them by the tracker. They exchange information about which pieces of the shared file they have, and may announce their *interest* in particular pieces held by the remote peer. The remote peer may then agree to send a particular piece to the peer. When a peer holds every piece, it can reconstruct a copy of the original file. It becomes a seeder, whose role is to continue sending pieces to leechers.

Periodically, peers will *update* the tracker to inform it of their progress on uploading and downloading the shared file. In return, the tracker responds with an updated list of peers in the swarm, allowing further new peers to join.

2.2 BitTorrent Protocol Messages

For peer/peer communication, the protocol specifies *messages* that can be exchanged between peers. We concern ourselves with the following relevant messages:

handshake. Sent immediately after a connection has been established between two peers; the peer that initiated the connection sends its handshake message first. Each peer includes a randomly-generated *peer ID* that the recipient uses to uniquely identify the sender.

extprotocol. Optionally sent after the handshake message, this message allows peers to exchange information about which protocol extensions they support.

bitfield. Sent after peers have exchanged handshake messages; the peer that initiated the connection sends its bitfield message first. The *bitfield* is a bit mask representation of the pieces that the sender claims to be holding; e.g., in a 10-piece torrent, the bitfield `1001010010` indicates that the peer holds pieces 0, 3, 5 and 8.

have. May be sent at any time during a connection's lifetime. Used to inform the recipient that the sender now holds a piece that the sender was not holding when the peers exchanged their bitfields; e.g., if peer P_A has the bitfield `1000` stored for peer P_B and P_B later sends the message have(2), P_A can update its bitfield for P_B to `1010`.

request. Requests that the recipient send a piece (or a block of a piece) that it has previously advertised.

piece. Contains the piece data that was requested by the recipient in an earlier request message.

keepalive. Idle peer/peer connections are usually closed after three minutes. This message is used to ask the recipient not to close the connection as a result of idleness, as the sender may send further messages later.

2.3 BitTorrent Indexing

The BitTorrent protocol does not specify how a torrent file should be circulated to other users interested in downloading the shared file. Consequently, *torrent indexing* web sites such as The Pirate Bay [21] were created to facilitate the organisation and distribution of torrent files. Many of them index copyright-infringing torrents — in the case of The Pirate Bay, this is its explicit purpose. The administrators of torrent indexing web sites are often targeted by legal action initiated by trade organisations representing copyright holders, who claim that online copyright infringement causes financial disruption to their members' businesses; these trade organisations have successfully persuaded courts in the United States, Sweden, Slovenia and other countries to order the closure of offending web sites and trackers.

3 Detecting Indirect Monitoring

A simple approach for performing indirect monitoring involves announcing to trackers and collecting the IP addresses of peers returned by the tracker. This technique offers a fast method of harvesting a large number of peers, but it has been shown by Piatek et al. [17] that IP address-based peer identification produces unreliable results. Furthermore, by announcing to trackers, monitors leave a trace of their presence: their IP addresses also appear in peer lists. We can then indirectly observe the behaviour of peers to identify differences between regular peers and monitors.

To motivate our subsequent work on direct monitoring, we first reassess techniques previously proposed to identify indirect monitors, and propose an additional novel feature for identifying them.

3.1 Methodology and Data Collection

To automatically collect information from BitTorrent trackers, we created our own indirect monitoring client that gathers newly-published torrent files from the Top 100 in each category on The Pirate Bay, and continually contacts each of the trackers and stores (IP address, port number, infohash, time) tuples from the peer lists that are returned; it then attempts to establish a TCP connection with each host and sends a handshake message to ensure that the host is in fact a BitTorrent peer. The monitor also requests from trackers the number of seeders and leechers in each swarm.

We collected data from July 21–28, 2009, routing our traffic through the Tor anonymity network [23]. This led to an excessive number of connections timing out or being dropped, so we collected data again without using Tor from August 4–6, 2009. A summary of data collected is presented in Table 1. The comparative success of the second trace when compared with the first seems to be entirely due to the poor performance of Tor.

Table 1. A summary of indirect monitoring activity.

	Jul 21–28, 2009	Aug 4–6, 2009
IP addresses seen	831,039	1,351,853
(IP, port) pairs seen	894,529	1,498,015
Torrents monitored	967	690
Trackers seen	196	181

3.2 Features for Detecting Monitors

Using this data we build profiles for the behaviour of BitTorrent clients, which we can use to differentiate regular peers from monitors. The assumption is that "anomalous" profiles may be indicative of the behaviour of monitors. To build such profiles, we first consider five features that have been previously proposed in the literature:

1. The proportion of a subnet that has been seen in BitTorrent swarms. Monitoring agencies may use a large proportion of their subnet for monitoring.
2. The length of time a peer spends in a swarm. Monitors may spend more time in the swarm than regular file-sharers.
3. The number of different (IP, port, infohash) combinations per IP address. Monitoring agencies may operate many clients from a single IP address.
4. Whether a peer reported by a tracker accepts incoming connections. Monitors may block all incoming connection attempts.
5. The number of swarms in which IP addresses from a particular subnet appear. Monitoring agencies may monitor many torrents from their subnet.

Features 1–4 have been suggested by Siganos et al. [20] and Piatek et al. [17], and feature 5 by Potharaju et al. [18]. Potharaju et al. also leverage web search engines to derive a database of the content being shared by each torrent, and look for peers that download multiple copies of the same content. Another potentially useful but untested feature is whether a peer is downloading content that is likely to appeal to very different audiences (e.g., a peer that downloads both classical and pop music tracks). We do not consider either of these features, as they cannot be calculated from information provided by trackers alone.

While investigating feature 4, we found that only 16% of peers in our datasets accepted incoming connections. Given the commonness of this behaviour, we conclude that the typical behaviour of a BitTorrent client is to reject incoming connection requests. This is likely due to BitTorrent users being affected by incorrectly-configured residential routers or firewalls. We show in Section 4 that many monitors *do* accept incoming connections, therefore we do not use this feature for detecting monitors.

Our heuristic for detecting monitors relies on the remaining four features. More precisely, we consider a peer likely to be a monitor if it appears in the top first percentile for each of the features (i.e., the highest number of connections,

the longest connection time, etc.); by applying this test we found 1,139 IP addresses that were in the top first percentile for all four features. To understand whether these features are effective at identifying monitors, we manually analysed these anomalies; they included IP addresses assigned to a company named Checktor [3], which offers commercial BitTorrent monitoring services, and 16 addresses assigned to a medium-sized computer security consultancy company that does not publicly acknowledge monitoring BitTorrent. Another subnet, which we saw in over 500 swarms, belongs to a company that advertises itself as providing "intellectual property advice", but does not specifically acknowledge monitoring BitTorrent. We also found two subnets assigned to hosting companies, one with IP addresses in 433 swarms and the other with IP addresses in 371 swarms. These hosting companies advertise themselves as providers of Internet services to businesses, rather than residential users, where BitTorrent traffic is more likely to be regulated. We speculate that copyright enforcement companies are using these hosting companies as a front to disguise their identities. We also identified a number of IP addresses allocated to large ISPs, such as Vodafone, Etisalat and SingNet. These ISPs have all been assigned very small subnets and therefore use NAT. Some of the 1,139 also seemed to be very active users on residential ISPs that were seeding a large number of files; while unusual, there was nothing to suggest that these peers were engaged in monitoring.

3.3 A Novel Feature

When comparing the profiles of suspicious peers that appeared to be monitoring with those that appeared to be subject to NAT, we noticed that the suspicious peers had multiple (IP, port) pairs in a number of different swarms. According to the BitTorrent protocol, a client should open a different port for each swarm that it joins; therefore, this behaviour is not expected from regular peers. While it would be possible for an (IP, port) pair to appear in more than one swarm, this should only happen when a peer has just left one swarm and joined another. The instances of peers in different swarms from ISPs that made heavy use of NAT, such as Vodafone and Etisalat, all had unique (IP, port) pairs. This observation led us to a new, sixth feature for identifying peers likely engaged in monitoring:

6. The number of times the same (IP, port) pair is observed concurrently in different swarms.

We considered any (IP, port) pair that appeared in four or more swarms to be suspicious. This feature found IP addresses assigned to Peer Media Technologies [16] (a well-known copyright enforcement agency) monitoring seven Harry Potter ebook and movie torrents, and the INRIA research institution [10], which had been overlooked by features 1–5 because so few torrents were being monitored, and because a very small proportion of INRIA's subnet was being used for monitoring. While we were collecting our data, INRIA did not publicly acknowledge monitoring BitTorrent; however, researchers there have since published work describing the detection of initial seeders of files [13].

3.4 Discussion

These results continue a line of work by Piatek et al. [17], Siganos et al. [20] and Potharaju et al. [18], who show that indirect monitoring of BitTorrent is occurring and can be detected by profiling specific characteristics of peers' behaviour.

The stopped message. The BitTorrent protocol allows a peer to send a stopped message in the announce to the tracker to inform it that the peer is leaving the swarm. The tracker should then remove the peer's IP address from its peer list. If a tracker correctly implements this rule of the protocol, an indirect monitor can send the message immediately after receiving a peer list and thus make itself undetectable. We tested a number of trackers' support for this message and while some trackers removed the IP address immediately, those operated by The Pirate Bay did not. By requesting from the tracker the number of completed downloads for each torrent, we found that The Pirate Bay balanced tracker load across six servers; it therefore seems probable that the two announces were being processed by different servers, which explains why peer IP addresses are not always removed from peer lists.

False positives and negatives. We note that, as a normal user of BitTorrent could be said to be "monitoring" the peers it connects to, it would be possible for a monitor to avoid detection by any set of features that tries to distinguish monitors from a regular peer. A monitoring client could avoid detection by our new feature by selecting a different port for each torrent, and monitoring agencies could use many different subnets and limit the amount of time that each IP address was used. This would make monitoring a much more expensive and time-consuming process, so while we cannot guarantee the detection of a monitor that deliberately tries to obscure its activities, we can detect monitors that try to maximise the number of file-sharers they find.

The suspicious behaviour we detected from the IP addresses of companies that acknowledge that they monitor BitTorrent (such as Checktor), and our detection of the INRIA monitors before they released their publication, does provide some ground truth to validate our methods. Inspecting our suspected monitors by hand, we found no results that appeared to be false positives (although we cannot absolutely rule out results that may be due to network behaviour we are unaware of). This suggests that our false positive rate is low. Inspecting a sample of the negative results, we did not find any that appeared to be monitors, although, for the reasons given above, it is harder for us to rule out false negatives.

We can make accurate comparisons between sets of features. Comparing the methods of Siganos et al., Piatek et al. and Potharaju et al. with our own, we found that they incorrectly identified IP addresses allocated to ISPs which make heavy use of NAT, such as Vodafone, Etisalat and SingNet. They also missed some of the smaller monitoring agencies such as Peer Media Technologies and INRIA. We can therefore be confident that the addition of our new feature decreases the false negative and false positive rate.

4 Detecting Direct Monitoring

Direct monitoring, in which monitors directly contact and probe other peers, was proposed by Bauer et al. as a method of improving the accuracy of file-sharing evidence collected by monitors [1]. However, it has not been shown conclusively that direct monitoring is being employed widely by copyright enforcement companies.

A direct monitor may operate *actively* (by announcing to the tracker, receiving peer lists and initiating outgoing connections to other peers), or *passively* (by placing itself into a swarm and listening for incoming connections only). Passive monitoring has the advantage of detecting peers using NAT and others that do not accept incoming connections; active monitoring can be performed more quickly and thus can monitor more peers across the same period. Initiating and listening for direct connections takes much longer than harvesting IP addresses from a tracker, so we concentrate on features that can be calculated without monitoring a large number of swarms.

4.1 Methodology and Data Collection

We created a number of customised BitTorrent clients, inserted them into swarms and observed their behaviour. Every protocol-compliant message sent to our clients was logged along with the timestamp, the message's payload, and the peer's IP address, port number and peer ID. As a side-effect of joining swarms, our clients regularly received peer lists from trackers after announcing to them, which we also stored for later use.

Table 2. A summary of direct monitoring activity.

	Aug 10–23, 2010	Feb 9–18, 2011	May 3–8, 2011
IP addresses seen	311,549	112,584	98,385
(IP, port) pairs seen	2,441,555	371,572	321,949
Torrents monitored	30	20	16
Trackers seen	20	12	12

We created two classes of clients: one designed to communicate with passive direct monitors (by harvesting peer lists and attempting to connect to each peer systematically), and another designed to communicate with active direct monitors (by joining the swarm and only listening for incoming connections). Since it is possible for monitors to engage in either or both forms of direct monitoring, this allowed us to determine which (if any) form is being used most frequently.

Our clients used three different bitfield-reporting strategies to detect discrepancies between the bitfields reported by other peers, so a peer intentionally misreporting its own bitfield would be noticeable:

Mirror strategy. Designed to appear as uninteresting as possible to other peers: reports to connecting clients that it holds the same pieces as the connecting client (by "mirroring" the client's bitfield and have messages back to them), does not send request messages for pieces of the shared file, and does not respond to request or piece messages.
Empty strategy. Appears to have joined the swarm recently: per the mirror strategy, but always reports an empty bitfield and does not mirror have messages.
Full strategy. Appears to be a seeder: per the mirror strategy, but always reports a full bitfield and does not mirror have messages.

Two groups of swarms were monitored: 6 sharing public domain files, and 60 sharing copyright-infringing files. Public domain torrents were sourced from ClearBits [4] and LinuxTracker [14]. Copyright-infringing torrents were selected from a range of categories on The Pirate Bay, including music, movies, TV shows, music videos and software. Torrents were selected from both within and outside of the Top 100. Table 2 summarises the data we collected.

4.2 Features for Detecting Monitors

We identify two features for distinguishing peers likely to be performing active monitoring:
Reported completion. Since our clients logged all bitfield messages, and most peers reconnected to our monitors, we could compare the bitfields the clients were sent and track their progression over time. Although the majority of peers reported steady progression towards completing the download, peers in 20 small subnets always reported completions of between 45% and 55%. For these IP addresses, further inspection of the bitfields showed no consistency: they appeared to be generated randomly, rather than reflecting a progressively completing download (compare Figures 3 and 4: black blocks indicate pieces of the file that a peer claims to have; white blocks are missing pieces). A peer that reports a piece as not downloaded when it had previously reported it as downloaded is lying about the parts of the shared file it is holding, and is therefore likely to be a monitor.
Connection frequency. It is common for peers to reconnect to peers they have discovered previously to check whether they are advertising new pieces that the peer still needs to download. Most peers connected to our clients over a 40-hour period during the entire monitoring period. However, 0.05% of the peer population, scattered across a low number of small subnets, connected to our monitors over a much longer 133-hour period; all of these peers were also detected by the "reported completion" feature. This is indicative of a group of peers more interested in analysing the download progress made by other peers rather than making any download progress of their own, and is another strong feature for identifying monitors.

Peers detected using these features superficially appeared to be active, but in fact they were not downloading the shared file; their IP addresses belong to subnets of three hosting companies. We can be sure that each connection was from the same BitTorrent client due to the unique peer ID in the handshake.

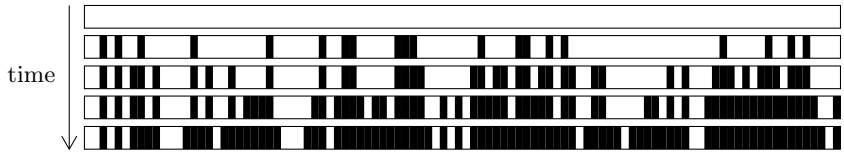

Fig. 3. The download progression of a regular peer. Its bitfield steadily progresses toward completion.

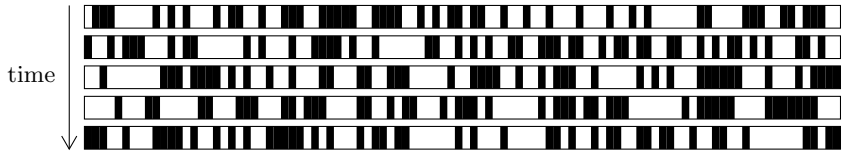

Fig. 4. The download progression of a monitoring peer. Its bitfield changes randomly over time.

This behaviour was not observed in any of the swarms sharing public domain content; the most likely explanation is that these were monitors. Notably, they did not request any pieces of the shared file after connecting, so it is questionable whether a copyright enforcement agency employing this technique could *prove* that other peers in the swarm were really sharing the file. We note that monitors could avoid detection by our "reported completion" feature by simulating a realistic bitfield over time, but establishing connections with other peers and then reporting a complete bitfield would be highly suspicious; additionally, as we could distribute monitors over several subnets, monitors could only avoid our "connection frequency" feature by making fewer connections, reducing their effectiveness.

We also experimented with several ineffective features; we briefly discuss them here, for the sake of completeness and to aid future research into direct monitoring:

Duration of connection. The protocol states that idle connections should be closed after 3 minutes to aid resource conservation. Peers may send keepalive messages to other peers to indicate that they wish to communicate again soon (e.g., to request a piece), and therefore want the connection to be kept alive. As there is no incentive for other peers to remain connected to our "mirror" and "empty" clients, it is expected that peers should spend little time connected to them, and conversely spend more time connected to our "full" clients; this was indeed the case, and we found no evidence that certain peers were deliberately keeping connections alive for monitoring purposes.

Protocol violations. All peers are expected to obey the protocol; e.g., if a peer advertises the availability of a piece, it should not request that piece in a future message. Similarly, a peer should not attempt to send a piece to another peer unless the receiving peer has explicitly requested it. Although we found no evidence that protocol violations indicate the presence of a monitor, instances of

protocol violation were observed from 4 IP addresses assigned to ISPs known to use NAT, indicating that this may instead be another suitable feature for identifying peers being subjected to NAT or firewalling.
Number of request messages sent. Since BitTorrent is a file-sharing protocol, it follows that peers should be expected to request pieces of the shared file from others; peers that do not request pieces of the file may therefore be participating in the swarm for reasons other than file-sharing (e.g., monitoring). However, a large proportion of peers (over 99.9%) connected to our clients without ever sending a request message for a piece of the file the clients were offering, and subsequently showed progress in downloading the file in future connections; therefore, this is an unlikely feature for detecting monitors.

4.3 Discussion

Table 3. ASes suspected of engaging in direct monitoring.

Number of Monitors	AS	Name
467	23504	Speakeasy, Inc.
202	174	Cogent/PSI
114	209	Qwest LLC
39	558	Net2EZ
17	27699	TELESP
17	1213	HEAnet Ltd

ASes involved in monitoring. Based on the features we identify, we suspect six ASes of harbouring a total of 856 peers engaging in direct monitoring (see Table 3). Two of these ASes (AS558 and AS1213) have previously been identified in the study by Potharaju et al. [18] as potential harbourers of monitoring agencies; we suspect a further four. AS209 was considerably more active in 2010 than in 2011; it may be that this AS was once being used by monitoring agencies, but no longer is.

Incidence of monitoring on The Pirate Bay. Our features only detected monitors in Top 100 torrents; this implies that copyright enforcement agencies are monitoring only the most popular content on public trackers. Movie and music torrents were most heavily monitored (by 65 and 26 monitors respectively), particularly by AS23504 and AS558; the other categories were less heavily monitored, although between 1 and 7 IP addresses suspected of monitoring were still present in each category.

The use of active vs. passive direct monitoring. All of the potential monitors we have identified engaged in active direct monitoring: our clients were unable to establish outgoing connections to them. This is understandable, as

monitors are able to communicate with many more peers (and therefore detect a larger number of downloaders) by harvesting peer lists and processing them systematically, as opposed to simply waiting for incoming connections for other peers.

Average time before monitors connect. 40% of the monitors that communicated with our clients made their initial connection within 3 hours of the client joining the swarm; the slowest monitor took 33 hours to make its first connection. The average time decreases for torrents appearing higher in the Top 100, implying that enforcement agencies allocate resources according to the popularity of the content they monitor.

Proportion of peers accepting incoming connections. The results of our 2009 study revealed that outgoing connections could only be made to 16% of peers. This fell to 7% in 2011. Since monitors currently engage in active direct monitoring only, peers may still be able to participate in a swarm undetected by enforcement agencies, who rely solely on a peer's ability to accept incoming connections in order to communicate with them.

False positives and negatives. As with indirect monitoring, the rate of false negatives is difficult to quantify, because a monitor can arbitrarily behave like a regular peer. However, this comes at the cost of a far-reduced monitoring capability. The more measures a monitor takes to increase its efficiency and coverage, the more easily it can be detected. As for false positives, the suspected monitors we found showed a highly irregular download progression (as shown in Figure 4); it is impossible for a peer sharing content to behave in this way, so we can be sure that they were not regular file-sharers. While we cannot be certain that they were monitors, it seems highly likely.

Some BitTorrent clients are known to deliberately misreport their bitfields when seeding, ostensibly to evade ISPs' traffic management policies that penalise BitTorrent seeders [22]: rather than sending a complete bitfield, these clients send a partially-complete bitfield and then immediately complete it with `have` messages for the pieces that were omitted (a technique named *"lazy bitfield"*); we note that this behaviour is now widespread among BitTorrent clients. Our customised clients eliminate this potential source of false positives by grouping the pieces advertised in a client's `bitfield` message with those advertised in `have` messages received in the subsequent 30 seconds as if they had all been advertised in the initial `bitfield` message.

To corroborate the potential sources of suspicious behaviour we had detected, we compared our results with the contents of public blocklists. These are lists of peers suspected of being involved in suspicious activity, and are typically created through manual analysis by a community of concerned users. We use such lists as a baseline for comparing our results and, in particular, for gaining an understanding of potential false positives and false negatives. More precisely, we used the Anti-Infringement blocklist available from I-BlockList [9], as it is popular among BitTorrent users.

As a preliminary step, we assessed the accuracy of the Anti-Infringement list by measuring the number of false positives it contained (i.e., the number of listed peers that are unlikely to engage in monitoring activity). To do so, we leveraged the observation that enforcement agencies have no incentive to monitor public domain torrents. Therefore, we consider an entry in the blocklist to be a false positive if we find a peer in the subnet engaged in the download of public domain torrents. During 27 days of monitoring, we found 5 false positives in the blocklist (out of 2,880 total subnets), and discarded them from the rest of this analysis. We considered the remaining 2,875 to be true positives (i.e., subnets that could contain monitoring peers).

Our direct monitoring analysis produced 593 peers (out of 856) that appear in subnets listed in the Anti-Infringement list. This represents a 69% overlap between our results and the contents of the list; therefore, the majority of our results are corroborated by the results of independent blocklists. In addition, our analysis identifies 263 peers (31% of our results) that, albeit displaying the same behaviour as monitoring peers (as determined by our detection features), do not currently appear in blocklists. We consider this a strong indication that these results are true positives of our analysis that are not detected by (manual) blocklisting techniques; BitTorrent users should therefore not rely solely on such speculative blocklists to protect their privacy, and should instead combine them with blocklists based on empirical research, such as those generated by Potharaju et al. [18], to reduce the number of false negatives encountered.

Finally, we measured the number of subnets in the Anti-Infringement list that were observed during direct monitoring and were *not* detected by our techniques; we consider peers in these subnets to be potential false negatives of our analysis that warrant further examination. We found 57 such peers. There are several reasons that these peers might not have been detected by our features: 53 disconnected from our monitoring clients at unexpected times, indicating possible network connectivity problems or malfunctioning BitTorrent clients. The remaining 4 used IP addresses whose ISPs are known to use NAT, potentially limiting their ability to communicate properly with our monitoring clients; these peers showed no signs of engaging in suspicious activity, so we suspect that their subnets were mistakenly added to the blocklist.

5 Conclusion

In this paper, we examined the current state of BitTorrent monitoring. We introduced several novel techniques for identifying peers that perform monitoring and validated them on large datasets. We determined that copyright enforcement agencies use indirect monitoring (confirming the results of earlier studies) as well as direct monitoring (a novel contribution of our work) to determine users' activity. From our experiments, we derived a number of interesting properties of monitoring, as it is currently performed: e.g., that monitoring is prevalent for popular content (i.e., the most popular torrents on The Pirate Bay) but absent for less popular content, and that peers sharing popular content are likely

to be monitored within three hours of joining a swarm. Finally, we found that publicly-available blocklists, used by privacy-conscious BitTorrent users to prevent contact with monitors, contain large incidences of false positives and false negatives, and recommended that blocklists based on empirical research [18] are used over speculative ones.

References

1. K. Bauer, D. McCoy, D. Grunwald, and D. Sicker. Bitstalker: Accurately and efficiently monitoring bittorrent traffic. In *Proceedings of the IEEE International Workshop on Information Forensics and Security (WIFS)*, London, UK, Dec. 2009.
2. Blizzard Entertainment. Networking help for the Blizzard Downloader, 2011. `http://us.blizzard.com/support/article.xml?locale=en_US&articleId=21077`.
3. Checktor. `http://www.checktor.com`.
4. ClearBits. `http://www.clearbits.net`.
5. G. Dán and N. Carlsson. Power-law Revisited: Large Scale Measurement Study of P2P Content Popularity. In *Proceedings of the International Workshop on Peer-To-Peer Systems (IPTPS)*, San Jose, CA, USA, 2010.
6. M. Freedman. Inaccurate Copyright Enforcement: Questionable "best" practices and BitTorrent specification flaws. Freedom to Tinker, 2009. `https://freedom-to-tinker.com/blog/mfreed/inaccurate-copyright-enforcement-questionable-best-practices-and-bittorrent-specificatio/`.
7. L. Guo, S. Chen, Z. Xiao, E. Tan, X. Ding, and X. Zhang. Measurements, Analysis, and Modeling of BitTorrent-like Systems. In *Proceedings of the USENIX Internet Measurement Conference (IMC)*, Berkeley, CA, USA, 2005.
8. J. Halliday. Filesharing prosecutions will face serious problems, says judge. The Guardian, 2008. `http://www.guardian.co.uk/technology/2011/feb/08/filesharing-prosecutions-digital-economy`.
9. I-BlockList. `http://iblocklist.com/lists.php`.
10. INRIA. `http://www.inria.fr/en/`.
11. R. Layton and P. Watters. Investigation into the extent of infringing content on BitTorrent networks. Technical report, Internet Commerce Security Laboratory, University of Ballarat, Australia, Apr. 2010.
12. S. Le Blond, A. Legout, F. Lefessant, and W. Dabbous. Angling for Big Fish in BitTorrent. Technical Report inria-00451282, INRIA, Sophia Antipolis, France, Jan. 2010.
13. S. Le Blond, A. Legout, F. Lefessant, W. Dabbous, and M. A. Kaafar. Spying the World from your Laptop — Identifying and Profiling Content Providers and Big Downloaders in BitTorrent. In *Proceedings of the USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET)*, San Jose, CA, USA, Apr. 2010.
14. LinuxTracker. `http://linuxtracker.org`.
15. Malibu Media, LLC v. John Does 1–5. Exhibit A to declaration of Tobias Fieser, 2012. `http://beckermanlegal.com/Lawyer_Copyright_Internet_Law/malibumedia_does1-5_120706OpposDeclarationFieserExA.pdf`.
16. Peer Media Technologies. `http://peermediatech.com`.
17. M. Piatek, T. Kohno, and A. Krishnamurthy. Challenges and Directions for Monitoring P2P File Sharing Networks — or — Why My Printer Received a DMCA Takedown Notice. In *Proceedings of the USENIX Workshop on Hot Topics in Security*, San Jose, CA, USA, 2008.

18. R. Potharaju, J. Seibert, S. Fahmy, and C. Nita-Rotaru. Omnify: Investigating the visibility and effectiveness of copyright monitors. In *Proceedings of the Passive and Active Measurement Conference (PAM)*, Atlanta, GA, USA, 2011.
19. Sandvine. Fall 2010 Global Internet Phenomena Report. `http://www.sandvine.com/downloads/documents/2010GlobalInternetPhenomenaReport.pdf`, 2010.
20. G. Siganos, J. Pujol, and P. Rodriguez. Monitoring the Bittorrent Monitors: A Bird's Eye View. In *Proceedings of the Passive and Active Measurement Conference (PAM)*, Seoul, South Korea, Apr. 2009.
21. The Pirate Bay. `http://www.thepiratebay.se`.
22. TheoryOrg. *BitTorrent Protocol Specification: bitfield*, July 2012. `http://wiki.theory.org/BitTorrentSpecification#bitfield:_.3Clen.3D0001.2BX.3E.3Cid.3D5.3E.3Cbitfield.3E`.
23. Tor Project. `https://www.torproject.org`.
24. M. Wojciechowski, M. Capotă, J. A. Pouwelse, and A. Iosup. BTWorld: towards observing the global BitTorrent file-sharing network. In *Proceedings of the ACM Workshop on Large-Scale System and Application Performance (LSAP)*, Chicago, IL, USA, 2010.
25. S. Wolchok and J. A. Halderman. Crawling BitTorrent DHTs for Fun and Profit. In *Proceedings of the USENIX Workshop on Offensive Technologies (WOOT)*, Washington, DC, USA, 2010.
26. B. Zhang, A. Iosup, J. Pouwelse, D. Epema, and H. Sips. Sampling Bias in BitTorrent Measurements. In *Proceedings of the European Conference on Parallel Processing (Euro-Par)*, Ischia, Italy, 2010.
27. C. Zhang, P. Dhungel, D. Wu, and K. W. Ross. Unraveling the BitTorrent Ecosystem. *IEEE Transactions on Parallel and Distributed Systems*, 22(7):1164–1177, July 2011.